Catherine’s notes on citation policies here: <https://docs.google.com/document/d/1onc6QXgFQSC1UxoZ6mqIq1ePGGyo8Opdm6DJNmI45Us/edit?hl=en_US>

Catherine’s notes on the citation classification literature: <https://docs.google.com/document/d/1kgIdQg5oKfEiYH4BBEcYQ_PKBgw-FvFPAfilc5uIlFs/edit?hl=en_US>

----------------------------------------------------------------------------------------------------------------------------

1/20/12

--------------------

**TO DO:**

- Look at the literature on citations (start with JASIS&T and ASIS&T conference, google for scholarly communication publishing venues, look at DataCite website, etc.)

- Read Howison 2011 CSCW paper and “Representing the Heavens”

--------------------

**MEETING NOTES:**

1) When citing software, what are we told to do and why?

* Look at policies in the APA handbook and other relevant handbooks to see how they want software cited (or if they even discuss it).
  + check this discussion: http://www.childrensmercy.org/stats/weblog2006/CitingSoftware.aspx
* Find a set of journals that are “usefully representative” of something and see what their policies about software are.

2) What do we actually do and why?

* Count the number of software mentions made in papers (from the “usefully representative” set of journals).
* Build a typology for how software is mentioned in papers.
  + Typology is emergent, but might include information such as...
    - Where the mention is in the paper. For example, in the natural sciences, a paper might have a separate “Methods and Materials” section where software (among other things) is discussed. A software mention in the body of the Methods section may also cite a paper that discusses the software, often the paper that originally introduced the piece of software.
      * Side note: Weird citation system where name of paper is not even listed in references. What’s up with that?
    - What version of software is used (if we can tell)
    - How that version was cited: Does it reference the original paper that unveiled the original version, or something else?
    - Why is this type of mention used?
  + Look at literature on citations to give us guidance on how to create typology
  + Look at DataCite, an organization trying to create standards for data citation. Might have information that is applicable re software.
* Knowing the typology and use of software citations, how would you communicate to people that you would like citations?

3) Other issues that came up in our discussion:

* Researchers are repurposing the citation system as a reputation system for software.
* Regarding how a version is cited: An possible issue is that a 2010 paper that used a 2009 version of a piece of software might only cite a 2000 paper introducing the software, which leaves out the contributions others have made to the software between 2000 and 2009.
  + Can we look at who made changes to the actual code used (if version number given) and whether they were an author on the cited paper?
  + Papers are more often published on new software than on improved existing versions of software, which leaves out the subsequent contributors to the software.
  + Even when papers are published on existing software, they may get published in lesser journals than those that publish papers unveiling new software. There may be a bias toward citing more prestigious journals over less prestigious journals.
    - can we look at this by finding the set of citable papers for a software project and seeing which are actually cited for that project?
  + There may be a bias toward citing the original paper anyway. (Maybe original paper is cited more, so bias toward more cited paper.)
  + Finally, a certain paper may be the “standard” paper that everyone is told to/expected to cite when referencing that software.
* How have researchers cited tools historically? The microscope was at one time a novel invention -- how would a tool like that be cited when first invented, and when would it stop being cited (when is it assumed that everyone knows you use a microscope?)?
  + Tools are lumped into the “Services” category in the scientific reputation economy. “Services” can include microscopes, software, office chairs, etc.
  + Trust in software and reliability. (Inverse relationship to how much you trust a piece of software and how much you feel you need to cite it in paper? Is it “research” only when it is new enough to have a potentially variable or adverse effect on your outcomes?)
* Do novice researchers cite software more than experts do? Is use of certain software assumed?
* Being “known” for tool-building is viewed negatively.

-------------------------------------------------------------------------------------------------------------------------

1/27/12

**TO DO:**

- James needs to think about what set of journals we’ll be looking at

- domain specific? (Probably biology, then)

- Catherine needs to look at software-citing policies\* (if any) of

- journals: START WITH Nature, THEN Nature Methods, Science, PNAS, Public Library of Science, Plus One, Cell

- style guides: (e.g., APA, MLA, Chicago, university dept websites?[Purdue])

- DataCite

-http://www.childrensmercy.org/stats/weblog2006/CitingSoftware.aspx

\* Specifically software, but also tools generally. More important than the specific rules is any text surrounding rules on what to do, why to do it that way, and when and when not to do it.

--------------------------------------

**MEETING NOTES:**

In future, we should ask paper authors why they made the citations they did (cited one paper and not another)

Catherine’s notes on the citation classification literature can be accessed here: <https://docs.google.com/document/d/1kgIdQg5oKfEiYH4BBEcYQ_PKBgw-FvFPAfilc5uIlFs/edit?hl=en_US>

Summary thoughts on citation taxonomies:

- Most classification systems don’t really examine tool use -- it will be relegated to maybe one category (“operational,” etc.). An exception was the Teufel paper that had two categories re tools, one for tool use and another for tool adaption or modification (potentially useful for our own taxonomy?)

- In the Cano paper, paper authors classified certain papers as both “conceptual” and “operational,” which may be an indication that methods/tools can also be seen as more than just functional.

---------------------------------------------------------------------------------------------------------------------------

2/3/12

**TO DO:**

- Go through journal articles and see how their data and software policies are actually carried out

- Try to use R package to do content analysis if possible

- Check out APA and MLA handbooks from the library to see their software policies

-------------------------------------

**MEETING NOTES:**

Catherine’s notes on citation policies here: <https://docs.google.com/document/d/1onc6QXgFQSC1UxoZ6mqIq1ePGGyo8Opdm6DJNmI45Us/edit?hl=en_US>

Content Analysis software: Atlas.ti, NVivo, CAQDAS, (there’s also an R package, not sure of the quality yet)

Software packages may include citation commands that will show you how the makers of the software want it to be cited (R does this)

Things to look for

- Mentions (“we used the ggplot2 package”)

- Simple mentions

- More descriptive mentions (if we found them)

- Footnotes

- Citations

- Code printouts in Methods and Materials

- Links

----------------------------------------------------------------------------------------------------------------------------

2/24/12

**TO DO:**

James:

* Follow-up email to Veronique re contact info/introductions for possible interviewees at other journals
* Contact Nature Structural and Molecular Biology re publishing a lab practice paper in their Correspondence section
* Form a Nature software archive?
* At some point, talk to Pat Galloway re digital preservation of software (how archived, accessible v. runnable, etc.)

Catherine:

* Look at protocol papers to see what they’re like
* Continue citation exploration, focusing on Nature Group journals (Nature, Nature Methods, and Nature Structural and Molecular Biology)
  + Find out how many citations are
    - Explicit about the version number
    - Located
    - Accessible
    - Runnable

---------------------------

**MEETING NOTES:**

Notes from call with Veronique Kierner here:

<https://docs.google.com/document/d/1CLMVeOuWALmjgfj3eEn0aD95zyngg4G0JV7unrNf5lY/edit>

----------------------------------------------------------------------------------------------------------------------------

3/5/12

**TO DO**

Catherine:

* pull together the big sample of 90 papers (via the method described below)
* ask life sciences librarian
  + what the best way to stratify the journals by impact factor would be
  + how the Web of Science database works if we want biology papers (we would like to cover all biology papers -- does simply choosing “biology” as a category cover cell biology, etc.?)
* Take a look at CAT (Coding Analysis Toolkit: <http://cat.ucsur.pitt.edu/app/main.aspx>) to see if this is something we can use instead of RQDA (it allows multiple users and calculates the agreement statistics between them, which is good)
  + See if we can upload the papers to CAT (might have to convert pdfs to txt)

<http://admin-apps.webofknowledge.com/JCR/JCR>

We want to choose by subject category, but we don’t know how the database is set up (choose biology? cell biology? Are they disjoint or overlapping? -- Need to ask a life sciences librarian)

-- once we figure out what categories to use, then we want to sort by impact factor and create three strata:

-- top ten

-- some middle 20 or so

-- the bottom 50 or so

We need a script that chooses years, months, or weeks

for each stratum

for each year (2000-2010)

for each week (1-52)

randomly choose a journal from that stratum (for top, 1-10 based on impact factor)

go into that journal and randomly choose an article (main article, not correspondence or review)

do this 30 times (x 3 strata = 90 articles)

james and I will code 20 together, than 35 each.

---------------------------------------------------

3/9/12

**TO DO:**

**MEETING NOTES:**

Discussion of sampling method issues here: <https://docs.google.com/document/d/1z2lbuZQAwvI-5za3vSsepdr2pcIf3AEkVwrRgvIGcvc/edit?pli=1>

---------------------

5/18/12

The previous citation coding schemes we’ve talked about:

Overarching citation goals:

Reproductibility / transparency

* Location / access / run
  + Direct citation: (URL, etc.)
  + Can you access it via a quick Google search

Credit-giving/blame-assigning:

* Who
  + Direct citation of person/group
  + How easy it is to find out who created this software (Google search, etc.)
* **Reproducibility and transparency.** How does this citation contribute to the ability to reproduce results?
* **Correctness.** How does this citation aid others in making sure the work done is correct? [RETHINKING THIS -- May not really be a function of citation)
* **Efficiency.** How does this citation allow for more researchers to consolidate, collaborate, or otherwise streamline work? Or, perhaps, how does this citation allow potential funders to identify work that should be supported?
* **Sustainability.**  How does this citation provide ways to motivate and undergird the continued production of research by scientists?
  + Does specifying how a piece of software should be cited (say, in the ReadMe file) help keep citations consistent?
    - Is there greater variance in mentions of software that does not give a clear indication of how it should be cited?
  + When we check to see if we can locate the software via the mention, we can see if there is a specification as to how to cite the software and then compare it to the way it is mentioned/cited in the paper.
* **Innovativeness.** How does this citation allow others from within or outside the field to use and contribute to that work in new ways?

Citation functions that help achieve some of the overarching goals:

* Explicit about the version number
* Located
* Accessible
* Runnable

Overarching goal is to see how much software is cited in science. We obviously can’t do that just by counting citations in journals, but we can follow up after we’ve gotten the preexisting citation info.

Two questions:

How do we cite software?

How effective is that citation practice for the goals of science?

How do authors decide how to mention software?

How do they decide whether to mention, say, software package or cite to paper?

Trying to figure out if mentions are consistent across package and time.

Does it vary by quality of journal (as identified by impact factor).

After initial phase, might be interesting to do a survey of the authors asking them what they think software citation practices should be, and then later interviews with them about what they did. However, important to avoid having authors connect the survey to interview, as their survey answers may prime them to answer interview questions in a certain way.

Does paying for software affect whether you would cite it or not?

Maybe set up a survey with hypotheticals that ask these types of questions.

--------

CODING SCHEME [NOT USING THIS SCHEME ANYMORE]:

Location of mention:

In-text

Reference section

Methods section [NOT USING THIS ANYMORE]

Other info:

Package name

Version number

URL

Date

Creator

**----------------------------------------------------**

5/21/12

Notes about coding:

References: For now we are going to do two levels of coding when it comes to references. The first time we encounter a reference, we will just code it as a reference and paste the full text into the ttl file.

The second step will be to automatically pull out all the references after that and resolve them to a URI and code them as a secondary item.

**CODING SCHEME (as of 5/21/12)**

in-text\_mention

software\_used

software\_not\_used

reference

package\_name

rdfs:label "whatever the name is"

version\_number

rdfs:label "whatever the version number is"

url

rdfs:label "whatever the version number is"

date

rdfs:label "whatever the version number is"

creator

rdfs:label "whatever the version number is"

Issues with “Software\_used\_not\_mentioned”:

“These papers are riven with artifacts created by software such as graphics, statistics, graphs. These types of software fall on a continuum of software that may or may not be important to the scientific community“

7, 26, 21, 25

----------------------------------------------------------

6/7/12

Over Skype. Went over questions Catherine had for James after initial coding. (Quoted from email, additional info from Skype conference in bold)

>> 1) Just to confirm, a software mention is only the first time a piece of software is mentioned in the text, correct?

>

> I'm glad you asked. No, I'd say each and every mention gets coded.

> **There was a question as to where each of the mentions ended. For example, pieces of software that have been mentioned multiple times in one paragraph could conceivably be one mention or many. It was decided that, rather than try to come up with one bright-line rule for dividing up mentions (by sentence, for example), we would try to divide them up semantically as best as we could. We said that we would revisit this after we had both finished the initial joint coding and compare notes.**

>>

>> 2) I have problems figuring out when to apply the package\_name code. I know we talked about this, and it's clear enough when we use the example of Microsoft Office, but I have more difficulty with the examples I encounter in the text. Basically I have been going to the software websites and seeing if they call it a "package" vs. a "program" or if they mention other programs with it (that might be part of a package?). It gets harder for me when they say things like "tool" or something else.

>

> Yeah, that's a tough one, I think any relevant identifier goes here. So we should really rename it "software\_name".

>

>>

>> 3) If the citation style of an article is such that the name of the paper author and date (e.g., "(Swofford 1998)") is given at the end of an in-text mention (as opposed to a number linking to it an item in the list of references), that doesn't count as an in-text mention of the "creator," does it?

>

> That's a tricky question. While I think that it does do a better job of crediting the authors you only get one or two names and we can't very well tell fields that use numbers that they are doing it wrong :) So, no, I don't think it does.

**Interesting issue that came up was whether software names provided just as much credit to the creator as pointing to a paper written by a creator. Perhaps this is only the case with well-known software?**

Catherine is going to look at Jena, an Apache project for processing RDF. General procedure is to take the file and load into data store (you can use an in-memory data store that Jena calls a “model”), and then query it using either SPARQL or programmatically in Jena by creating Java objects that correspond to the RDF subject-predicate-object triples. Catherine will attempt to output the number of software mentions in her ttl file and a unique list of software names coded. The Jena help email list is a helpful resource.

James will do his initial coding, then email Catherine for her file before running the joint statistics. Catherine will edit her file to reflect the above discussed changes before sending it to James. Another Skype session is set up for Tuesday, June 19, 2:30pm CST.

-------------------

6/19/12

Skype meeting.

Created new google doc of the coding scheme and notes about the codes: <https://docs.google.com/document/d/1ZUb6nk26SssRF7dRmRKjPduinczivk8WoZXi2Ev4cTo/edit>

Use configuration details

If “version 2003” then coded with both version\_number and date

Code is sometimes the best embodiment of the model

- Put in a note about coding models as computer programs in certain instances

If I want to change the coding scheme, drop it into the Code notes google doc at the bottom.

James mentioned a time when he saw an Acknowledgements where they thanked people who contributed yeast strains, but not software.

---------------

7/26/12

CATHERINE TO DO:

- Create html file that, for each article, displays a column of Catherine codes and a column of James codes.

* Store each article in an array
* For each article, store a list of codes for that article in an array
* Print to html by looping through the arrays

Possible way to compare my and James coding schema:

Levenshtein difference (Java has way to do this - in Apache string utils)

- Shows the edit difference

- Go with longest common subsequence (don’t allow substitution)

- Ultimately going to hold off on this method -- maybe will add similarity metrics after initial processing

----------------------------

8/29/12

To Do for the semester:

- For each journal in the sample, look up the citation policies for that journal

- What should be cited

- Do they have a specific data policy, software policy, archiving data/software policy (for citation formatting/policy)?

- Interview authors of the articles (at least 10, but not realistic all 90) to ask them about

- what software they used

- how they cited and what they cited

- NOTE: We need to make sure that we are not making it sound like they should have done something, we’re just interested in their practices

- Look at references and find out if you can

- actually find these pieces of software

- Write paper by early next semester

-----

Comparing James and Catherine codes:

- Article has\_selection and then a new url that has, for example, “j01”

- NOTE: There are some articles in the ttl files that are not in the sample, so when querying it, make sure to check if it is in the subsample.

- We changed the way we coded so we didn’t have as many anonymous nodes

bioj:has\_selection bioj:2004-40-NAT\_GENET-C02 .

bioj:2004-40-NAT\_GENET-C01 rdf:type bioj:selection ;

- Although

---------------

8/30/12

**Discussion of Agreement Statistics**

We are doing 3 different agreement analyses:

1) Can we agree over seeing software mentions?

* Counts of mentions
* Or do Cohen’s Kappa on sentences in paper
* Column names:
  + Number identified - C
  + Number identified - J
  + C not J
  + J not C
  + Matching
  + Non-Matching
  + Total Coded

2) When we do, can we classify correctly?

* Cohen’s Kappa over number of software mentions

3) When we use “software\_name” and “version\_number” codes, do we have agreement on the rdf label?

* For each agreement, is it identical or non-identical?

--------------

9/5/12

TO DO:

-Write-up the coding process and agreement process

- Calculate Cohen’s Kappa using IRR package in R (or whatever’s easiest) for each code

- Make a dataframe for each code

URI James Catherine

C01 0 1

- Calculate percentage of how much our codable units match up (our matches over our matches plus our bioj:nones)

---------------------------------

10/25/12

James gave overview of project to Clay:

Collaborative work of software

- Instrumenting software collections (where software will report its use more)

- Look at how software is mentioned in the science literature

Clay and Ken have been working on an IARPA project

- Creating a model to explain how science works

- Identifying via NLP emerging research in scientific literature

- Definition of emergence was something they struggled with

- Their working definition is based on if an entity is mentioned at a particular time and not mentioned previously, that would constitute “emergence.” (Clarify with Clay - I think it involved dividing up the graph into discrete blocks of time?)

- Current project: classifying entities in scientific literature

- instruments

- concepts

Hal Dumais at U of Maryland

Michael Collins (created Collins Parser)

Kathy something at Columbia

Louis Gravano, Columbia - metadata

Drago ? at U of Michigan - citation analysis

? - sentiment analysis

Simonet ? does work on citations (check again with Clay - there was a paper by this person that was pertinent to citation stuff we were doing?)

(Get List from Clay later)

Where our projects intersect:

- Use codes from our sample to train their system to recognize software (one of the entities they’re interested in tracking)

\* ( CG post-meeting note: our “software\_mentions” are counted semantically, rather than as word- or phrase-level instances. May require some additional processing before transition to their system.)

- Could we eventually compare the features the system finds important to the features humans find important (our code being proxy for humans)?

- James: It would be interesting to see how a piece of software is used in different contexts, etc., or perhaps moves across communities/connected to research.

- Does this piece of software drive the emergence of new questions?

- Phenomenon of interest: how Bayesian statistics have seeped into every single field now.

If you cite the paper, there’s no way to put a version number.

- With typical references (reference to a paper), not only is the thing you are referring to crystallized, but the actual thing you are referencing (information found in paper) is found specifically where you have pointed. With software paper references, however, chances are that the “thing” you are referencing is not found in the paper (indirect/inefficient locator [paper → software] , possibly inaccurate/untrustworthy locator [new version, etc.]

Questions that were brought up:

- If the producer has cited a software paper. (Are producers more likely to cite software than non-producers?)

- What are the configuration\_details? Are they science parameters or technical parameters?

(- Catherine Note: Does a lack of configuration\_details (or other details) indicate that the piece of software is seen as less important by authors, or just so ubiquitous that it doesn’t need mentioning?)

- In a field (like NLP) that is more tool-driven, how are the tools cited?

Future Work:

- Compare biology software mentions with NLP software mentions

- Talking to Matt could be helpful

- Train new coders

----------------

06/26/12

**Research Questions for the Software Citation Study**

How ought we mention and cite software in the literature?

What proportion of the literature is mentioning software?

How do they mention it?

Taxonomy of mention types (in-text, footnote, URL, citation to primary science paper, citation to software publication, citation to package, source code tag)

How does it function as a credit-giving system?

How effective are the mentions at functions of citation?

* Identification
* Location (we could find x% of the code, y% we have decent confidence we could find exact version)
* Reproducibility (what proportion of the papers provided links to code? 0%, I think? Or enough configuration info that we could confidently find the functions used)
* Credit-giving (what proportion of mentions, and what mention type)

**Coding references within articles**

We discussed how to code each bioj:has\_reference selection for the type of item it leads to, whether it identifies creators and whether it contains a url. The idea is to apply the same codes as would be used in the rest of the article, and also code as to document type.

For now we will begin with the agreement sets.

* Find each bioj:has\_reference
  + If the selection is another article,
    - Check to see whether it is in the Mendeley set yet, if not
      * Find it and add it to the Mendeley database
      * Give it a file name and tag in the style of “Author-Year” (add authors until it is unique)
  + Create an identifier for the referenced item in the form “bioj-cited:Author-Year”
    - If an article, local name should match the file name
    - If a url, project page, manual, name it something sensible!
  + Apply a code indicating the item the selection refers to. Possible (but not exhaustive codes) include:
    - citec:domain\_publication
    - citec:software\_publication
    - citec:project\_page
    - citec:users\_guide
    - citec:project\_name
  + Apply creator, url, version, date, or any other codes from the original list, as applicable

An example of a completed coded selection follows:

bioj-cited:1993-McCune rdf:type bioj:selection ;

bioj:full\_quote """MCCUNE, B. 1993. Multivariate analysis on the PC-ORD system. Oregon State University, Corvallis, Oregon. USA. http:// bioag.byu.edu /zoology/crandall\_lab"""" # 1993-McCune tag on paper in Mendeley ;

ca:isTargetOf [ rdf:type ca:CodeApplication ;

ca:hasCoder "jhowison" ;

ca:appliesCode citec:software\_publication ; #domain\_publication, #users\_guide, #project\_page, #project\_name. Go and look at the type of thing that is at the 'end' of the reference.

] ;

ca:isTargetOf [ rdf:type ca:CodeApplication ;

ca:hasCoder "jhowison" ;

ca:appliesCode [ rdf:type citec:creater ;

rdfs:label "McCune, B" ]

] ;

ca:isTargetOf [ rdf:type ca:CodeApplication ;

ca:appliesCode [ rdf:type citec:url ;

rdfs:label "http:// bioag.byu.edu /zoology/crandall\_lab" ] ;

] ;

* What to do when a single reference has more than one in-text mention?
  + For bioj:2003-39-J\_COMPUT\_AID\_MOL\_DES-B04 I gave the bioj:has\_reference the same bioj-cited value, but did not redo the reference coding

---------------

Fri Aug 30 08:10:53 CDT 2013

---------------

The citation project.

We have a random sample of 90 articles, and we've coded them.

**Paper one:** We can say how people mention software and assess how good a job these mentions do with each of the citation functions.

1. What are those citation functions (review that literature)

I keep thinking that the locating/credit-giving taxonomy comes from the literature, but I can't find it in our notes. So perhaps it's something that we cooked up. Basically citations originally formed a locating function and have come to form a reputation system (see literature on impact factor issues etc). After location comes one of either a) review, b) replication, c) modification (Guttman scale, ie if c=yes, then a and b must be Yes). For literature the fixed form means that finding the article is relatively easy. For software you struggle to find the specific version (a), see if it still works (b) and may have rights/datedness issues for modification (c).

2. Create a coding scheme (or a query scheme)

- For locating this might involve trying to find the software (and the specific version).

- For crediting this might involve just seeing if the mention has a creator.

- Looking at the packages themselves, how many appear to still be actively developed? How many have software publications? How many are commercial?

3. Review the citation policies of the journals in the sample.

4. Look at how the software publications (if any) line up with the version date of the software, are there any unacknowledged contributors?

Immediate TODOs:

1. Find and document the citation policies of the journals in the set.

2. Build out and test the coding scheme for citation functions

Finding and using (Guttman Scale)

Identification (present because of mention)

Location mode - through request?

Location mode - URL following? Actually work?

Location mode - web search. Able to find it?

Is specific version available? Or only the package?

Location - specific version

Access - availability (binary only)

Access - availability (source code)

Replication - specific version still runs

Modification - code can be modified (rights specified, if none specified then assume no).

Mention is the map, what is found at the end of it?

Protocol. Action and questions about what was found.

Action 1: Identify software

Q: is there a name?

Q: is there a URL?

Q: is there a creator name?

Q: is there a version number?

Q: is there a date?

Q: is there an invitation to request the software?

Action 2: Locate the software.

2.1: Follow URL (if url)

2.2: Search using information in the mention (name/creator/version/date)

Question: Does the URL still work?

Have I found the package?

Have I found the specific version mentioned in the paper (if any?)

If no version mentioned, can I find the version around the date?

If a specific version or historical version near date:

Is source available?

For latest version

Is source available?

Is there a binary available?

Do the licensings terms indicate modifiability (if no license then no).

Credit-giving

Creator mentioned (in paper, either in the text or in the reference)

If publication, date difference between version and publication date

If publication and date difference, evidence of additional contributors?

**Paper two:** We can interview all of the paper authors (starting with some subset, numbers 4,9 and 13 in each strata), asking:

- For a narrative of the paper's construction

- What other software might have been used (giving us percentage of citation figures, albeit biased, since those that mentioned some software probably are predisposed to mentioning software.)

- How these authors think about the citation of software

- Where they store the software/workflow for those papers.

**Paper three:** Work with Ken towards machine learning based on this coding set.